TODO: incluir portada oficial

Categories in Homotopy Type Theory

(Título provisional - busca uno definitivo - probablemente en mayo)

Pedro Bonilla Nadal

Pedro Bonilla Nadal *Categories in Homotopy Type Theory*

.

Master's thesis.
Academic year 2020-2021.

Thesis supervisor	Dr. Ángel González Prieto *Departamento - Universidad*	Máster en Matemáticas y Aplicaciones Universidad Autónoma de Madrid

Contents

Part I

Category Theory

Chapter 1

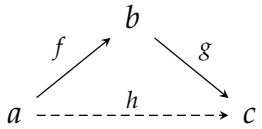
First Notions of Category Theory

> "The human mind has never invented a labor-saving machine equal to algebra."
>
> *Stephan Banach (1925)*

For us, Categories are an area of interest on its own rights, rather than merely an elegant tool. In this sense we will introduce this theory with a personal point of view, trying to emphasize the intuitive ideas behind each notion.

Thus, in addition to the formal content, we will also provide enough information so that an avid reader gain quick intuitive understanding of the subject. With this objective in mind, we will get into the habit of introducing the first examples even before introducing the formal definitions. Thus, when the formal definition is introduced, it becomes more meaningful and helps the intuition to understand the concept.

The fundamental idea of Category Theory is that a many properties can be unified if expressed with diagram and arrows. Intuitively, a diagram is a directed graph, such that each way of going from a node to another are equals. For example, in the diagram:



Means that $f \circ g = h$. This approach to mathematics emphasises at the relationships between elements, rather than at the elements themselves (and from them derive their relationships). In general, dashed lines means that the existence of that particular arrow is uniquely determined by the solid arrow presents in the diagram.

In this first chapter we will introduce the notion of category and their first properties. The principal references for this chapter are [1] and [2].

1.1 Metacategories

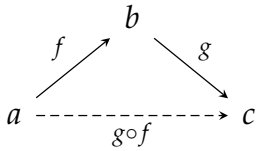
We will start by defining a concept independent of the set theory axioms: the concept of *metacategory*. Once this is done we will follow reinterpreting this definitions in the context of set theory. We follow [1] for these definitions.

Traditionally, mathematics is based on the set theory. When we start set theory it is not necessary (it is not possible) to define what a set is. It is similar with the concepts of element and belonging, which are basic to set theory. Category theory can also be used to found mathematics. In this sense we will give definitions based on other concepts such as object, arrow, composition.

Definition 1.1.1. A *metagraph* consist of *objects*: $a, b, c..$ and *arrows* $f, g, h....$ There are also two pairings: dom and codom. This pairings assigns each arrow with an object. An arrow f with $\mathsf{dom}(f) = a$ and $\mathsf{codom}(f) = b$ is usually denoted as $f : a \to b$.

Definition 1.1.2. A metacategory is a metagraph with two additional operations:

- *Identity*: assigns to each object a an arrow $1_a : a \to a$.
- *Composition*: assigns to each pair of arrows f, g with $\mathsf{codom}(f) = \mathsf{dom}(g)$ and arrow $g \circ f$ such that the diagram:



  commutes. The arrow $g \circ f$ is called the *composite* of f and g.

There are additional two properties:

- *Associative*: given arrows f, g, h, we have that,

$$(g \circ f) \circ h = g \circ (f \circ h).$$

- *Unit*: given an object a, and arrows f, g such that $\mathsf{dom}(f) = a$ and $(g) = a$, we have that,

$$1_a \circ g = g, \qquad f \circ 1_a = f.$$

In the context of a metacategory, arrows are often called *morphisms*.

We have just define what a metacategory is without any need of set and elements. In most cases we will rely in a set theory interpretation of this definitions, as most examples will rely on this theory. Nonetheless, whenever possible, we will define the concepts working only in terms of objects and arrows, having therefore the theory as independent as possible to this theory.

1.2 ZFC Axioms

(Maybe) TODO: Introduce fundamental terminology and problems related to classical set theory.

Once we have talk so much about this we may also formally introduce the stuff. Is not so long (?) - ok, maybe it is. I am letting this as a TODO and continue with other stuff until this is more done.

Further on i left a detailed TODO of things from this section that are referenced in the text.

Detailed TODO:

- small sets and large sets.

1.3 Set theory categories

Further on we will work based on set theory.

Definition 1.3.1. A category (resp. graph) is an interpretation of a metacategory (resp. metagraph) within set theory.

That is, a graph/category is a pair (O, A) where a collection O consisting of all objects as well as a collection A consisting of all arrows. Their elements holds the same properties that objects and arrows satisfy on metacategories / metagraph.

We will focus on the category. We can also define the function homeset of a category $C = (O, A)$, wrote as hom_C, as the function:

$$\begin{aligned} \mathsf{hom}_C : O \times O &\mapsto \mathcal{P}(A) \\ (a, b) &\mapsto \{f \in A | f : a \to b\} \end{aligned}$$

When there is no possibility of confusion we will state $c \in C$ without specifying . We will refer to the collection of objects of a category C as $Ob(C)$ and the collection of arrows as $Ar(C)$.

Definition 1.3.2. We say that a category is *small* if the collection of objects is given by a set (instead of a proper class). We say that a category is *locally small* if every homesets is a set.

We proceed to introduce a comprehensive list of examples, so that it is already introduced in subsequent chapters.

Example 1.3.1. [1]

- The elementary category:

[1] Debería cambiar este entorno por una pequeña subsección?

  - The category $0 = (\emptyset, \emptyset)$ where every property is trivially satisfy.
  - The category $1 = (\{e\}, \{1_e\})$.
  - The category $2 = (\{a, b\}, \{1_a, 1_b, f : a \to b\})$

- Discrete categories: are categories where every arrow is an identity arrow. This are sets regarded as categories, in the following sense: every discrete category $C = (A, \{1_a : a \in A\})$ is fully identified by its set of object.

- Monoids and Groups: A monoid is a category with one object (regarding the monoid of the arrows). In the same way, if we requires the arrows to satisfy the inverse property, we can see a group as a single-object category.

- Preorder: From a preorder $(A, \leq)$ we can define a category $C = (A, B)$ where B has an arrow $e : a \to b$ for every $a, b \in A$ such that $a \leq B$. The identity arrow is the arrow that arise from the reflexive property of the preorders.

- Large categories: these categories has a large set of objects. For example:

  - The category *Top* that has as objects all small topological spaces and as arrow continuous mappings.
  - The category *Set* that has as objects all small sets and as arrows all functions between them. We can also consider the category Set_* of pointed small sets (sets with a distinguish point), and function between them that maps one distinguish point into another.
  - The category *Vect* That has as object all small vector spaces and as arrows all linear functions.

- The category *Hask* of all Haskell types and all possible functions between two types.

- Include more examples as categories are used in the text.

Note that, for example, as natural numbers can be seen as either a set or a preorder, they also can be seen as a discrete category or a preorder category.

1.3.1 Properties

We proceed considering some elementary properties on categories.

We can see that is common in mathematics to have an object of study (propositional logic clauses, groups, Banach spaces or types in Haskell). Once the purpose of the study of these particular set of objects is fixed, it is also common to proceed to consider the transformations between these objects (partial truth assignments, homomorphisms, linear bounded functionals or functions in Haskell).

In categories, we have a kind of different approach to the subject. Instead of focusing on the objects themselves, we focus on how they relate to each other.

That is, we focus on the study of the arrows and how they composes. Therefore we can consider equal two objects that has the same relations with other objects. This inspire the next definition:

Definition 1.3.3 (Definition 1.1.9 [2])**.** Given a category $C = (O, A)$, a morphism $f : a \to b \in A$ has a *left inverse* (resp. *right inverse*) if there exists a $g : b \to a \in A$ such that $g \circ f = 1_b$ (resp. $f \circ g = 1_a$). A morpishm is an isomorphism if it has both left and right inverse, called the *inverse*. Two object are isomorphic is there exists an isomorphism between them.

Is easy to follow that this functions define bijections between $\mathsf{hom}(a, c)$ and $hom(b, c)$ for all $c \in O$. Also one can see that if a morphism has a left and a right inverse, they must be the same, thus implying the uniqueness of the inverse.

We will now proceed to talk about certain arrows and objects that have properties that distinguish them from others. To accompany and gain an intuition of these properties, I find the most useful example is that of the *FinSet* category, of all finite sets and functions between them. Considering special arrows:

Definition 1.3.4. An arrow f is a *monic* (resp. *epic*) if it is left-cancelable (resp. right-cancelable), i.e. $f \circ g = f \circ h \implies g = h$ (resp. $g \circ f = h \circ f \implies g = h$).

In FinSet this arrows are the injective functions (resp. surjective). Considering special objects:

Definition 1.3.5. an object a is *terminal* (resp. *initial*) if for every object b there exists an unique arrow $f : b \to a$ (resp. $f : a \to b$). An object that is both terminal and initial is called *zero*.

In FinSet this objects is the initial object is the empty set and the terminal object the one point set. In the Pointed

Proposition 1.3.1. Every two terminal object are isomorphic.

Proof. Every terminal object has only one arrow from itself to itself, and necessarily this arrow has to be the identity. Let a, b be terminal object and $f : a \to b$ and $g : b \to a$ be the only arrows with that domain and codomain. Then $f \circ g : a \to a \implies f \circ g = 1_a$. Analogously $g \circ f = 1_b$.

□

Another important property in category theory is the *duality property*. In sort, this property tell us that for every theorem that we prove for categories, there exist another theorem that is automatically also true. To prove this, we define the concept of *opposite category*.

Definition 1.3.6 (Definition 1.2.1 [2])**.** Let C be any category. The opposite category C^{op} has:

- the same objects as in C,

- an arrow $f^{op} \in C^{op}$ for each arrow $f \in C$, so that the domain of f^{op} is defined as the codomain of f and viceversa.

The remaining structure of the category C^{op} is given as follows:

- For each object a, the arrow 1_a^{op} serves as its identity in C^{op}.
- We observe that f^{op} and g^{op} are composable when f and g are, and define $f^{op} \circ g^{op} = (g \circ f)^{op}$

The intuition is that we have the same category, only that all arrows are turned around. We can see that for each theorem T that we prove, we have reinterpretation that theorem to the opposite category. Intuitively this theorem is an equal theorem in which all the arrows have been turned around. For example: the proposition 1.3.1 can be reworked as:

Proposition 1.3.2. Every two initial object are isomorphic.

That is because being initial is the dual property of being terminal, that is, if $f \in C$ is a terminal object then $f^{op} \in C^{op}$ is an initial object. Being isomorphic is its own dual.

1.3.2 Transformation in categories

This is one of the main ways of defining a category: consider a collection of objects and the standard way of transforms one into each other. Then, we may also follow our study defining the structure preserving transformation of categories.

Definition 1.3.7. Given two categories C, B, a *functor* $F : B \to C$ is a pair of functions $F = (F' : Ob(C) \to Ob(B), F'' : Ar(C) \to Ar(B))$ (the *object function* and the *arrow functor* respectively) in such a way that:

$$F''(1c) = 1_{F'c},\ \forall c \in Ob(C); \qquad F''(f \circ g) = F''f \circ F''g,\ \forall f, g \in Ar(C).$$

That is, a functor is a morphism of categories. When there is no ambiguity we will represent both F' and F'' with a single symbol F acting on both objects and arrows. Also, as you can see in the definition, whenever possible the parentheses of the functor will be dropped. This loss of parentheses will be replicated thorough the text, whenever possible. Lets provide some examples:

Example 1.3.2.

- Forgetfull functor:
- Poincaré functor:
- Čech:
- Group actions:
- `Maybe` Method in Haskell:

An important consideration on functors is that a functor F can be defined only pointing out how it map arrows, as how F maps object can be defined with how it map identity arrows.

We can now construct the category of all small categories $\mathcal{Cat}$. This category has as object all small categories and as arrows all functor between them. Note that $\mathcal{Cat}$ does not contain itself.

We can consider some properties in functors. Before defining then, note that we can consider a functor $T : C \to D$ as a function over homeset, that is, a function:

$$T : \hom_C(a, b) \to \{Tf : Ta \to Tb | f \in C\} \subset \hom_D(Ta, Tb)$$

Definition 1.3.8. A functor $T : C \to D$ is *full* if it is surjective as a function over homesets, i.e. if the function $T : \hom_C(a, b) \to \hom_D(Ta, Tb)$ is surjective for every $a, b \in C$. A functor is *faithfull* if it is injective over homesets.

As we have defined the concept of opposite category, we can consider functors $T : C^{op} \to D$. This functor, is called a *contravariant* functor from C to D, in oposition of a covariant functor from C to D. A functor $T : C \to D^{op}$ is usually called a *covariant* functor from

We shall continue defining natural transformation. In the words of Saunders Mac Lane:

> Category has been defined in order to define functor and functor has been defined in order to define natural transformation.

One can see a functor $T : C \to D$ as representation of a category in another, in the sense that a functor provide a picture of the category C in D. Further elaborating into this idea, we can consider how to transform these drawings into each other.

Definition 1.3.9. Given two functor $T, S : C \to D$, a *natural transformation* $\tau : T \to S$ is a function from $Ob(C)$ to $Ar(D)$ such that for every arrow $f : c \to c' \in C$ the following diagram:

$$\begin{array}{ccc} Tc & \xrightarrow{\tau c} & Sc \\ \downarrow{\scriptstyle Tf} & & \downarrow{\scriptstyle Sf} \\ Tc' & \xrightarrow{\tau c'} & Sc' \end{array}$$

commutes. A natural transformation where every τc is invertible is called a *natural equivalence* and the functors are *naturally isomorphic*.

That is a natural transformation is a map between pictures of C into D. Note that a natural transformation acts only on the domain of objects. Lets provide some examples

Example 1.3.3.

- The bidual space:
- A polymorphic function in Haskell:

1.3.3 Some constructions

In this last subsection we will introduce some standard construction on categories, along with some examples of these constructions.

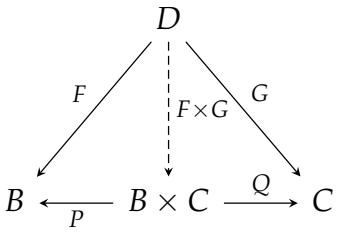
Product Category

We present now one of the most usual construction in mathematics: the product. We will consider the "universal" properties of product on the next chapter. By now, we present the product of categories.

Definition 1.3.10. Let B, C be categories. Then the *product category* $B \times C$ is the category that has as objects the pairs $\{< b, c >: b \in Ob(B), c \in Ob(C)\}$ and as arrows the pairs of arrows $\{< f, g >: f \in Ar(B), g \in Ar(C)\}$. The composition of arrows if defined by the elementwise composition.

It is clear that we can define to functor $P : B \times C \to B$ and $Q : B \times C \to C$ that restricts the category to each of its component parts (functorial axioms follow immediately). Moreover, we can see that any functor $F : D \to B \times C$ will be uniquely identified by its composition by P and Q.

Complementary, for any two functors $F : D \to B, G : D \to C$ we can define an functor $F \times G : D \to B \times C$ that apply $(F \times G) < f, g >=< Fg, Gg >$. Expressed as a diagram:



A functor $F : B^{op} \times C \to D$ is called *bifunctors*. Arguably the most important bifunctor is the hom_C function seen as a functor. Given a category C we can see $\mathsf{hom}_C : C^{op} \times C \to Set$ as a bifunctor such that:

$$\mathsf{hom}_C < a, b >= \mathsf{hom}_C(a, b) \qquad \forall a, b \in Ob(C)$$

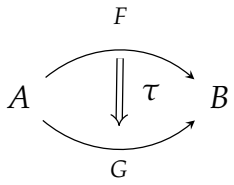
for the object. For the arrows:

$$\begin{aligned} \mathsf{hom}_C < f^{op}, g >: \mathsf{hom}_C(a', b) \to \mathsf{hom}_C(a, b') \qquad & \forall f : a \to a', g : b \to b' \in C \\ \mathsf{hom}_C < f^{op}, g > (h) = g \circ h \circ f \qquad & \forall h \in \mathsf{hom}_C(a', b) \end{aligned}$$

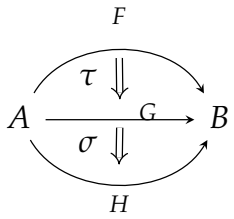
Functor Categories

We continue defining functor categories, that is, categories where we consider the functors as objects and natural transformation as arrows in some sense. This concept will be instrumental in further consideration in the realm of functional programming (in particular, in the definition of monad).

Before explaining composition between natural transformation, it is useful to present the following diagram. Let B, C be categories, $F, G : B \to C$ be functors and $\tau : F \to G$ natural transformation τ, it is common to represent this structure with:



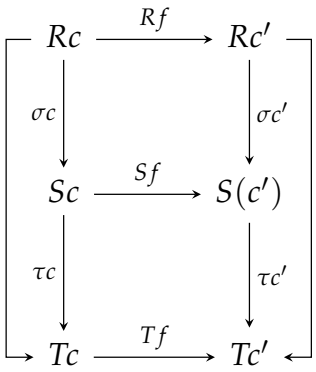
Lets define the composition of two natural transformation. We will start defining the composition of two natural tranformations σ, τ as in:



Due to this representation, this composition of natural transformations is called *vertical composition*, in opposition to the *horizontal composition* (def. 1.3.13).

Definition 1.3.11. Let C and B be two categories, $R, S, T : C \to B$ be functors, and let $\tau : R \to S$, $\sigma : S \to T$, we define the composition $(\tau \circ \sigma)c = \tau c \circ \sigma c$.

To see that $(\tau \circ \sigma)$ is a natural transformation it suffices the following diagram[3]:



Definition 1.3.12. Let B, C be categories. We define the category B^C as the functor category from B to C, that is, the category with functors from B to C as object, natural transformation as arrows, and composition as defined in 1.3.11.

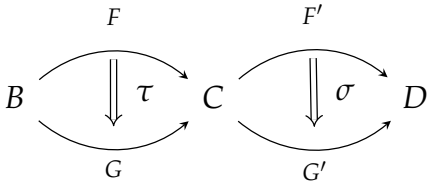
we now present some examples to provide some intuition about when this type of construction are considered.

Example 1.3.4.

- The category of group action over a set.

- C^2 also called the arrow category.
- Relation with the product category.
- Study composition of polymorphic function, and put that in either this example or in the next one.

Note that this is not the only way in which to define the composition of natural tranformation. In fact, we can define another functor category. In this case we compose two natural transformation as in:



Lets formalize this composition:

Definition 1.3.13. Let B, C, D be categories, $F, G : B \to C, F', G' : C \to D$ be functors, and let $\tau : F \to G$, $\sigma : F' \to G'$, we define the composition $(\tau \circ \sigma) : F \circ F' \to G \circ G'$ such that

$$(\sigma \circ \tau)c = Fc \circ G'\tau c$$

for all $c \in B$.

With this horizontal composition we can properly defined.In this case we can see that the composition of two natural transformation is indeed a natural transformation due to the commutativity of:

$$\begin{array}{ccc} F'Fc & \xrightarrow{\sigma Fc} & G'Fc \\ \downarrow{\scriptstyle F'\tau c} & \searrow^{(\sigma\circ\tau)c} & \downarrow{\scriptstyle G'\tau c} \\ F'Gc & \xrightarrow{\sigma Gc'} & G'Gc' \end{array}$$

With this definition of composition we can consider another different category: the category of all functors of all (small)[2] categories, that is, the category that has all functors as object, and has the natural transformation with horizontal composition as arrows.

When we have to consider both compositions at the same time we denote the vertical composition with $\tau \cdot \sigma$ and horizontal composition with $\tau \circ \sigma$, as in [1]. Lastly we have to consider how this composition relate to each other. This is seen in the *interchange law*:

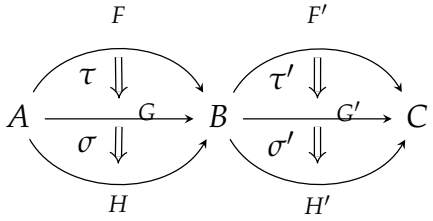
Proposition 1.3.3. Let A, B, C be categories, $F, G, H : A \to B$, $F', G', H' : B \to C$ be functors and $\tau : F \to G, \sigma : G \to H, \tau' : F' \to G', \sigma' : G' \to H'$ be natural

[2] me parece que tengo que incluir esto pero lo hago más por miedo que por conocimiento

transformations, then:

$$(\sigma' \circ \sigma) \cdot (\tau' \circ \tau) = (\sigma' \cdot \tau') \circ (\sigma \cdot \tau)$$

Proof. We have a structure like



From the naturality of τ' we have that for all $c \in A$:

$$\begin{aligned}
((\sigma' \cdot \tau') \circ (\sigma \cdot \tau))(c) &= H'((\sigma \cdot \tau)c) \circ (\sigma' \cdot \tau')(Fc) \\
&= H'(\sigma c \circ \tau c) \circ (\sigma'(Fc) \circ \tau'(Fc)) \\
&= H'(\sigma c) \circ H'(\tau c) \circ \sigma'(Fc) \circ \tau'(Fc) \\
&= H'(\sigma c) \circ \sigma' G c \circ G' \tau c \circ \tau'(Fc) \\
&= (\sigma' \circ \sigma)(c) \circ (\tau \circ \tau)(c) \\
&= ((\sigma' \circ \sigma) \cdot (\tau' \circ \tau))(c).
\end{aligned}$$

□

Comma Category

Chapter 2

Unnamed chapter

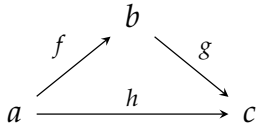
2.1 Definition

2.2 Universality

In this section we present the concept of universality. This concept is behind lots of mathematical properties. Intuitively, universality is an efficient way of expressing an one-to-one correspondence between arrows of different categories. This one-to-one relationship is usually expressed via "given an arrow y it exists one and only one arrow x such that <insert your favorite universality property".

Prior to the formal definition, we shall introduce an example. Probably the first contact that any mathematician has with universality is when we first try to define a function $f : \mathbb{R} \to \mathbb{R}^2$. We quickly understand that defining such a function is equivalent to define two $g, h : \mathbb{R} \to \mathbb{R}$(we further explain the product in 2.2.2). Another example is given when you consider the space quotient of a set A for a relationship over it. In this case, giving a function from $A/$ is the same as giving a function from A that maintains the equivalence relationship $a\ b \implies f(a) = f(b)$. A similar concept lays for almost every quotient structure. Here is the general concept.

Definition 2.2.1. If $S : D \to C$ is a functor and c an object of C, an *universal arrow* at from c to S is a pair $< r, u >$ with $r \in D, u \in Ar(C)$, such that the diagram:



commutes.

2.2.1 Yoneda's lemma

Yoneda's lemma is one of the main results of category theory. This results is due to japanese professor Nobuo Yoneda. We know about Yoneda's life thanks to the elegy that was written by Yoshiki Kinoshita[4]. Yoneda was born in Japan in 1930, and received his doctorate in mathematics from Tokyo University in

1952. He was a reviewer for international mathematical journals. In addition to his contributions to the field of mathematics, he also devoted his research to computer science.

Mac Lane[1] assures the lemma first appeared in his private communication with Yoneda in 1954. With time, this result has became one of the most relevant. The idea behind the Yoneda's lemma is better understood in the context of Moduli problems.

Statement and proof

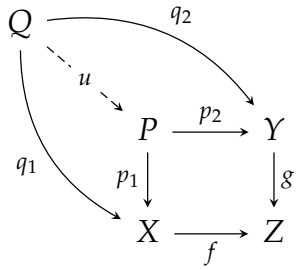
The Yoneda's Embedding

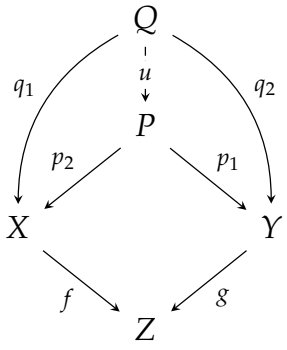
We can easily see that

2.2.2 Some properties expressed in terms of universality

We can see that the Yoneda lemma provide an embedding form

- product
- pullback: dejo estos dibujos muy convenientes guardados.





2.3 Adjoints

- Def
- Note that every adjuntion raise a universal arrow

2.4 Monad

Bibliography

[1] Saunders Mac Lane. *Categories for the working mathematician*. Vol. 5. Springer Science & Business Media, 2013.

[2] Emily Riehl. *Category theory in context*. Courier Dover Publications, 2017.

[3] Sigur. *Using Tikz-cd to Illustrate Composition of Natural Transformations*. Computer Science Stack Overflow. URL: https://tex.stackexchange.com/questions/505322/using-tikz-cd-to-path-to-illustrate-composition-of-natural-transformations%20(version:%202021-02-05).

[4] Kinoshita Yoshiki. "Prof. Nobuo Yoneda passed away". In: (1996). URL: https://www.mta.ca/~cat-dist/catlist/1999/yoneda.